

Sparsity-accuracy trade-off in MKL

Ryota Tomioka & Taiji Suzuki*
 {tomioka, t-suzuki}@mist.i.u-tokyo.ac.jp

Abstract

We empirically investigate the best trade-off between sparse and uniformly-weighted multiple kernel learning (MKL) using the elastic-net regularization on real and simulated datasets. We find that the best trade-off parameter depends not only on the sparsity of the true kernel-weight spectrum but also on the linear dependence among kernels and the number of samples.

1 Introduction

Sparse multiple kernel learning (MKL; see [9, 12, 2]) is often outperformed by the simple uniformly-weighted MKL in terms of accuracy [3, 8]. However the sparsity offered by the sparse MKL is helpful in understanding which feature is useful and can also save a lot of computation in practice. In this paper we investigate this trade-off between the sparsity and accuracy using an elastic-net type regularization term which is a smooth interpolation between the sparse (ℓ_1 -) MKL and the uniformly-weighted MKL. In addition, we extend the recently proposed SpicyMKL algorithm [15] for efficient optimization in the proposed elastic-net regularized MKL framework. Based on real and simulated MKL problems with more than 1000 kernels, we show that:

1. Sparse MKL indeed suffers from poor accuracy when the number of samples is small.
2. As the number of samples grows larger, the difference in the accuracy between sparse MKL and uniformly-weighted MKL becomes smaller.
3. Often the best accuracy is obtained in between the sparse and uniformly-weighted MKL. This can be explained by the dependence among candidate kernels having neighboring kernel parameter values.

*Both authors contributed equality to this work.

2 Method

Let us assume that we are provided with M reproducing kernel Hilbert spaces (RKHSs) equipped with kernel functions $k_m: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ ($m = 1, \dots, M$) and the task is to learn a classifier from N training examples $\{(x_i, y_i)\}_{i=1}^N$, where $x_i \in \mathcal{X}$ and $y_i \in \{-1, +1\}$ ($i = 1, \dots, N$). We formulate this problem into the following minimization problem:

$$\begin{aligned} & \underset{\substack{f_m \in \mathcal{H}_m \\ (m=1, \dots, M), \\ b \in \mathbb{R}}}{\text{minimize}} \quad \sum_{i=1}^N \ell \left(\sum_{m=1}^M f_m(x_i) + b, y_i \right) + C \sum_{m=1}^M \left((1-\lambda) \|f_m\|_{\mathcal{H}_m} + \frac{\lambda}{2} \|f_m\|_{\mathcal{H}_m}^2 \right), \end{aligned} \quad (1)$$

where in the first term, f_m is a member of the m -th RKHS \mathcal{H}_m , b is a bias term, and ℓ is a loss function; in this paper we use the logistic loss function. The second term is a regularization term and is a mixture of ℓ_1 - and ℓ_2 - regularization terms. The constant C (> 0) determines the overall trade-off between the loss term and the regularization terms. Here the first regularization term is the linear sum of RKHS norms, which is known to make only few f_m 's non-zero (i.e., sparse, see [16, 18, 1]); the second regularization term is the squared sum of RKHS norms. The two regularization terms are balanced by the constant λ ($0 \leq \lambda \leq 1$); $\lambda = 0$ corresponds to sparse (ℓ_1 -) MKL and $\lambda = 1$ corresponds to uniformly-weighted MKL.

Due to the representer theorem (see [13]), the solution of the above minimization problem (1) takes the form $f_m(x) = \sum_{i=1}^N k_m(x, x_i) \alpha_{i,m}$ ($m = 1, \dots, M$); therefore we can equivalently solve the following finite-dimensional minimization problem:

$$\begin{aligned} & \underset{\substack{\alpha_m \in \mathbb{R}^N \\ (m=1, \dots, M), \\ b \in \mathbb{R}}}{\text{minimize}} \quad L \left(\sum_{m=1}^M \mathbf{K}_m \alpha_m + b \mathbf{1} \right) + C \sum_{m=1}^M \left((1-\lambda) \|\alpha_m\|_{\mathbf{K}_m} + \frac{\lambda}{2} \|\alpha_m\|_{\mathbf{K}_m}^2 \right), \end{aligned} \quad (2)$$

where $\mathbf{K}_m \in \mathbb{R}^{N \times N}$ is the m -th Gram matrix, $\alpha_m = (\alpha_{1,m}, \dots, \alpha_{N,m})^\top$ is the weight vector for the m -th kernel, and $\mathbf{1} \in \mathbb{R}^N$ is a vector of all one; in addition, $L(\mathbf{z}) = \sum_{i=1}^N \ell(z_i, y_i)$. Moreover, we define $\|\alpha_m\|_{\mathbf{K}_m} = \sqrt{\alpha_m^\top \mathbf{K}_m \alpha_m}$.

The minimization problem (1) is connected to the commonly used “learning the kernel-weights” formulation of MKL in the following way. First let us define $g(x) = (1-\lambda)\sqrt{x} + \frac{\lambda}{2}x$ for $x \geq 0$ and $g(x) = -\infty$ for $x < 0$. Since g is a concave function, it can be linearly upper-bounded as $g(x) \leq xy - g^*(y)$, where $g^*(y)$ is the concave conjugate of $g(x)$. Thus substituting $x = \|\alpha_m\|_{\mathbf{K}_m}^2$ and $y = \frac{1}{2\beta_m}$ for $m = 1, \dots, M$ in Eq. (2), we have:

$$\underset{\alpha_m, b, \beta_m}{\text{minimize}} \quad L \left(\sum_{m=1}^M \mathbf{K}_m \alpha_m + b \mathbf{1} \right) + C \sum_{m=1}^M \left(\frac{\|\alpha_m\|_{\mathbf{K}_m}^2}{2\beta_m} - g^* \left(\frac{1}{2\beta_m} \right) \right),$$

where

$$g^*\left(\frac{1}{2\beta_m}\right) = -\frac{1}{2} \frac{(1-\lambda)^2 \beta_m}{1-\lambda\beta_m}.$$

Minimizing the above expression wrt α_m while keeping the loss term unchanged (i.e., $\sum_{m=1}^M \mathbf{K}_m \alpha_m = \mathbf{z}$ for some \mathbf{z}), we have $\alpha_m = \beta_m \alpha^*$ and finally we can rewrite Eq. (2) as follows:

$$\underset{\alpha^* \in \mathbb{R}^n, b \in \mathbb{R}, \beta \in \mathbb{R}^M}{\text{minimize}} \quad L(\mathbf{K}(\beta) \alpha^* + b \mathbf{1}) + \frac{C}{2} \left(\alpha^{*\top} \mathbf{K}(\beta) \alpha^* + \sum_{m=1}^M \tilde{g}(\beta_m) \right),$$

where $\mathbf{K}(\beta) = \sum_{m=1}^M \beta_m \mathbf{K}_m$ and $\tilde{g}(\beta_m) = -2g^*(1/(2\beta_m))$. Therefore Eq. (2) is equivalent to learning the decision function with a combined kernel $\mathbf{K}(\beta)$ with the Tikhonov regularization on the kernel weights β_m . Note that $\tilde{g}(\beta) = \beta$ (ℓ_1 -MKL) if $\lambda = 0$ and $\tilde{g}(\beta)$ approaches the indicator function of the closed interval $[0, 1]$ in the limit $\lambda \rightarrow 1$ (uniformly-weighted MKL). In this paper we call $\beta = (\beta_m)_{m=1}^M$ a *kernel-weight spectrum*.

The regularization in Eq. (1) is known as the elastic-net regularization [19]. In the context of MKL, Shawe-Taylor [14] proposed a similar approach that uses the square of the linear sum of norms in Eq. (2). Both Shawe-Taylor’s and our approach use mixed (ℓ_1 - and ℓ_2 -) regularization on the *weight vector* (or its non-parametric version) in the hope of curing the over-sparseness of ℓ_1 -MKL.

There are alternative approaches that apply non- ℓ_1 -regularization on the *kernel weights* β_m . Longworth and Gales [11] used a combination of ℓ_1 -norm constraint and ℓ_2 -norm penalization on the kernel weights. Kloft *et al.* [8] proposed to regularize the ℓ_p -norm of the kernel weights (see also [4]). Our approach (and [11]) differ from [8] in that we can obtain different levels of *sparsity* for all $\lambda < 1$ (see bottom row of Fig. 1), whereas for all $p > 1$ the resulting kernel-weight spectrum is dense in [8]. Note also that uniformly-weighted MKL ($\varphi = \infty$ in [11] and $p = \infty$ in [8]) corresponds to $\lambda = 1$ in our approach, which may be a possible advantage of our approach.

3 Results

3.1 Real data

We computed 1,760 kernel functions on 10 binary image classification problems (between every combinations of “anchor”, “ant”, “cannon”, “chair”, and “cup”) from Caltech 101 dataset [5]. The kernel functions were constructed as combinations of the following four factors in the preprocessing pipeline:

- Four types of SIFT features, namely hsvsift (adaptive scale), sift (adaptive scale), sift (scale fixed to 4px), sift (scale fixed to 8px). We used the implementation by van de Sande *et al.* [17]. The local features were sampled uniformly (grid) from each input image. We randomly choosed

200 local features and assigned visual words to every local features using these 200 points as cluster centers.

- Local histograms obtained by partitioning the image into rectangular cells of the same size in a hierarchical manner; i.e., level-0 partitioning has 1 cell (whole image) level-1 partitioning has 4 cells and level-2 partitioning has 16 cells. From each cell we computed a kernel function by measuring the similarity of the two local feature histograms computed in the same cell from two images. In addition, the spatial-pyramid kernel [7, 10], which combines these kernels by exponentially decaying weights, was computed. In total, we used 22 kernels (=one level-0 kernel + four level-1 kernels + 16 level-2 kernels + one spatial-pyramid kernel). See also [6] for a similar approach.
- Two kernel functions (similarity measures). We used the Gaussian kernel:

$$k(q(x), q(x')) = \exp\left(-\sum_{j=1}^n \frac{(q_j(x) - q_j(x'))^2}{2\gamma^2}\right),$$

for 10 band-width parameters (γ 's) linearly spaced between 0.1 and 5 and the χ^2 -kernel:

$$k(q(x), q(x')) = \exp\left(-\gamma^2 \sum_{j=1}^n \frac{(q_j(x) - q_j(x'))^2}{(q_j(x) + q_j(x'))}\right)$$

for 10 band-width parameters (γ 's) linearly spaced between 0.1 and 10, where $q(x), q(x') \in \mathbb{N}_+^n$ are the histograms computed in some region of two images x and x' .

The combination of 4 sift features, 22 spacial regions, 2 kernel functions, and 10 parameters resulted in 1,760 kernel functions in total.

Figure 1 shows the average classification accuracy and the number of active kernels obtained at different values of the trade-off parameter λ . We can see that sparse MKL ($\lambda = 0$) can be significantly outperformed by simple uniformly-weight MKL ($\lambda = 1$) when the number of samples (N) is small. As the number of samples grows the difference between the two cases decreases. Moreover, the best accuracy is obtained at more and more sparse solutions as the number of samples grows larger.

3.2 Simulated data

In order to explain the results from the image-classification dataset in a simple setting, we generated three toy problems. In the first problem we placed one Gaussian kernel over each input variable that was independently sampled from the standard normal distribution. The number of input variables was 100. We call this setting *Feature selection*. In the second problem we increased the

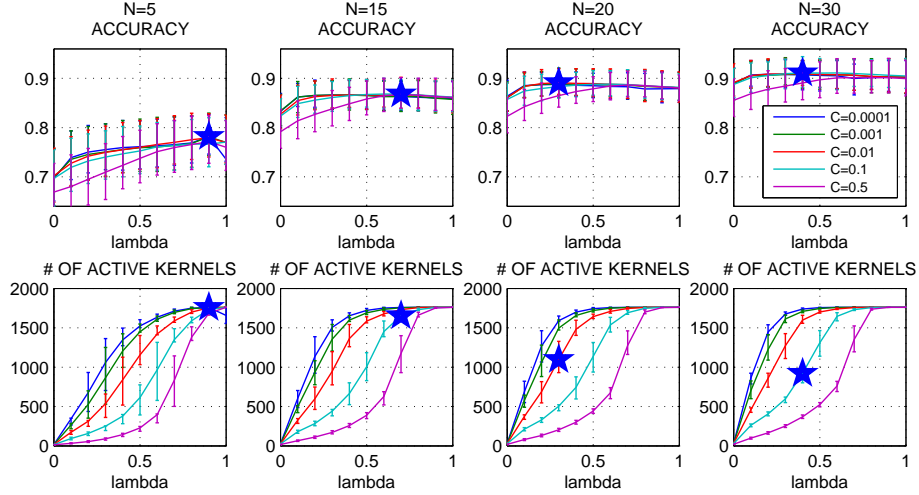


Figure 1: Image classification results from Caltech 101 dataset. The trade-off parameters λ that achieve the highest test accuracy are marked by stars.

variety of kernels by introducing 12 kernels with different band-widths on each input variable. The number of input variables was 10. We call this setting *Feature & Parameter selection*. In the third problem, we used the same 12 kernel functions with different band-widths but *jointly* over the same set of 10 input variables. We call this setting *Parameter selection*. The true kernel-weight spectrum $(\beta_m)_{m=1}^M$ was changed from sparse (only two non-zero β_m 's), medium-dense (exponentially decaying spectrum) to dense (uniform spectrum).

Figure 2 shows the test classification accuracy obtained from training the proposed elastic-net MKL model to nine toy-problems with different goals and different true kernel-weight spectra. We choose the best regularization constant C for each plot. First we can observe that when the goal is to choose a subset of kernels from *independent* data-sources (top row), the best trade-off parameter λ is mostly determined by the true kernel-weight spectrum; i.e., small λ for sparse and large λ for dense spectrum. Remarkably the sparse MKL ($\lambda = 0$) performs well even when the number of samples is smaller than that of kernels if the true kernel-weight spectrum is sparse. On the other hand, if we also consider the selection of kernel parameter through MKL (middle row), the best trade-off parameter λ is often obtained in between zero and one and seems to depend less on the true kernel-weight spectrum. This finding seems to be consistent with the observation in [19] that the elastic-net ($0 < \lambda < 1$) performs well when the input variables are linearly dependent because kernels that only differ in the band-width can have significant dependency to each other. Furthermore, if we consider the selection of kernel parameter only (bottom row), the accuracy becomes almost flat for all λ regardless of the true kernel-weight spectrum. The

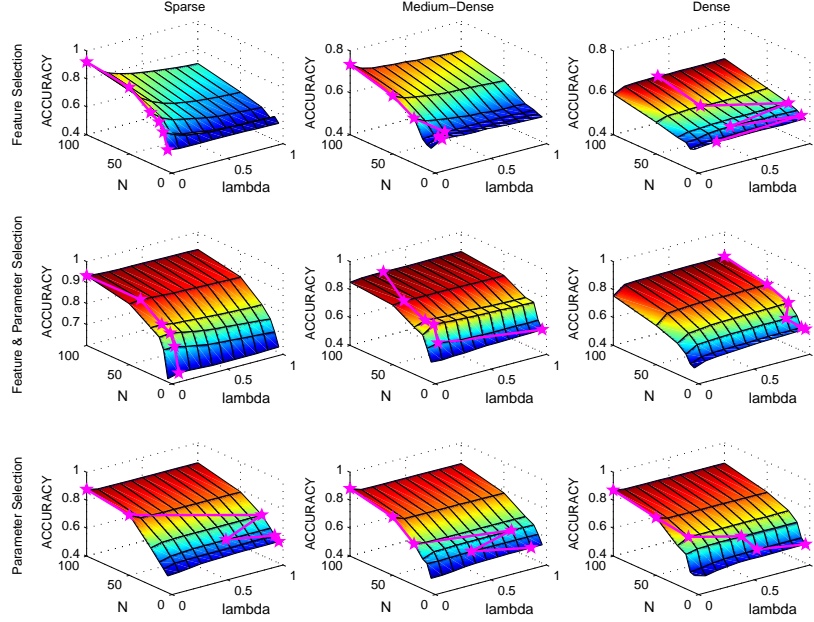


Figure 2: Classification accuracy obtained from the simulated datasets. The magenta colored curves with stars denote the value of trade-off parameters λ that yield the highest test accuracy.

behaviour in the Caltech dataset seems to be most similar to the second column of the second row (feature & parameter selection under medium sparsity).

4 Summary

In this paper, we have empirically investigated the trade-off between sparse and uniformly-weighted MKL using the elastic-net type regularization term for MKL. The sparsity of the solution is modulated by changing the trade-off parameter λ . We consistently found that, (a) often the uniformly-weighted MKL ($\lambda = 1$) outperforms sparse MKL ($\lambda = 0$); (b) the difference between the two cases decreases as the number of samples increases; (c) when the input kernels are independent, the sparse MKL seems to be favorable if the true kernel-weight spectrum is not too dense; (d) when the input kernels are linearly dependent (e.g., kernels with neighboring parameter values are included), intermediate λ value seems to be favorable. We have also observed that as the number of samples increases the sparser solution (small λ) is preferred. It was also observed (results not shown) that sparser solution is preferred when the noise in the training labels is small.

References

- [1] F. Bach. Consistency of the group lasso and multiple kernel learning. *JMLR*, 9:1179–1225, 2008.
- [2] F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *the 21st International Conference on Machine Learning*, pages 41–48, 2004.
- [3] C. Cortes. Can learning kernels help performance? Invited talk at International Conference on Machine Learning (ICML 2009). Montréal, Canada, 2009.
- [4] C. Cortes, M. Mohri, and A. Rostamizadeh. L2 regularization for learning kernels. In *Proc. UAI 2009*, June 2009.
- [5] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *IEEE. CVPR 2004 Workshop on Generative-Model Based Vision*, 2004.
- [6] P. Gehler and S. Nowozin. Let the kernel figure it out; principled learning of pre-processing for kernel classifiers. In *IEEE CVPR 2009*, 2009.
- [7] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *JMLR*, 8:725–760, 2007.
- [8] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien. Efficient and accurate lp-norm multiple kernel learning. In *Advances in NIPS 22*. 2010.
- [9] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the Kernel Matrix with Semidefinite Programming. *J. Machine Learning Research*, 5:27–72, 2004.
- [10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE CVPR*, volume 2, pages 2169–2178, 2006.
- [11] C. Longworth and M. Gales. Combining derivative and parametric kernels for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(4):748–757, 2009.
- [12] C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *J. Mach. Learn. Res.*, 6:1099–1125, 2005.
- [13] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, 2002.

- [14] J. Shawe-Taylor. Kernel learning for novelty detection. In NIPS 08 Workshop: Kernel Learning – Automatic Selection of Optimal Kernels, 2008.
- [15] T. Suzuki and R. Tomioka. SpicyMKL. Technical Report arXiv:0909.5026, 2009.
- [16] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc. B*, 58(1):267–288, 1996.
- [17] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (in press), 2010.
- [18] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. Roy. Stat. Soc. B*, 68(1):49–67, 2006.
- [19] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B(Statistical Methodology)*, 67(2):301–320, 2005.